REMARKS/ARGUMENTS

Applicant respectfully requests the consideration of the following remarks and the reconsideration of the present application.

Claims 1-66 were rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,446,198 (hereinafter "Sazegari"). Applicant respectfully disagrees.

The Office Action asserted that

> "Fig. 3-4 and column 4, lines 5-32, specifically disclose that a single permute instruction may perform simultaneous lookups in multiple sub-tables. The contents of the sub-tables (data1 and data2) may be stored in multiple vector registers of the CPU. The permute instruction then operates on (performs lookups into) the two registers (sub-tables), as show in Fig. 3-4." (Item 42 on page 10, the Office Action of May 9, 2005).

Applicant respectfully disagrees. Sazegari explicitly shows that:

> "... the permute instruction can be used to perform 16 simultaneous lookup operations on a 32-byte entry table." (Col., 4, lines 24-26, Sazegari)

It is clear that Sazegari describes the simultaneous lookups in a ***single*** table. Sazegari does not suggest simultaneous lookups in multiple sub-tables. In Sazegari, each index in the permute mask is for indexing into the single 32-entry table; and each of the lookups is for looking up into the single 32-entry table. In Sazegari, the input from the two registers form a single lookup table. Sazegari does not describe or show lookups in multiple sub-tables. It is improper to reject based on the speculation of the existence of lookups in multiple sub-tables. Thus, the rejection of claims 1-66 under 35 U.S.C. 102(e) is improper.

Thus, at least for the above reasons, Sazegari does not anticipate the pending claims.

Further, new claims 67-69 recite additional limitations that are not in Sazegari. For example, claim 67 recites:

> 67. (New) A method as in claim 1 wherein the plurality of segments of bits in the string of bits are of variable lengths.

In contrary, the indices in the permute mask vector as shown in Sazegari are of equal lengths (5 bits). Further, for example, claims 68 and 69 recites:

> 68. (New) A method as in claim 1 wherein each of the plurality of indices corresponds to a different one of the plurality of look-up tables.
>
> 69. (New) A method as in claim 1 wherein each of the plurality of look-up tables is larger than a vector register.

Further, for example, claim 2 recites "receiving a plurality of data elements specifying the plurality of segments in the string of bits." Note that claim 1 recites "generating a plurality of indices using a plurality of segments of bits in the string of bits". In Sazegari, when the permute instruction is executed, the 6th – 8th bits of a byte of the index register 36 is not used. These bits cannot be considered as corresponding to the data elements that specify the plurality of segments of bits to generate the indices, since they are not used to specify and generate the indices for the lookups in executing the permute instruction. In fact, the indices in the permute mask are at the predetermined locations (e.g., the first five bits of each byte in the permute mask). There would be no need to receive data elements to specify the plurality of segments of the permute mask. Thus, the rejection for claim 2 is improper.

Further, for example, claim 3 recites "the microprocessor is a media processor integrated with a ***memory controller*** on a single integrated circuit". Sazegari does not show the integration of a media processor with a ***memory controller*** on a single integrated circuit. Thus, the rejection for claim 3 is improper.

Further, for example, claim 5 recites "receiving a bit pointer, wherein the plurality of segments in the string of bits are determined using the bit pointer and the plurality of data elements". Note that claim 1 recites "generating a plurality of indices using a plurality of segments of bits in the string of bits". The description of Sazegari (Col. 4, lines 63-67) is about a different instruction, the selection instruction. The mask for the selection instruction does not determine how the indices are partitioned in the permute mask for the permute instruction. In fact, the indices in the permute mask are at the predetermined locations (e.g., the first five bits of each byte in the permute mask). There would be no need to receive a bit pointer to determine the plurality of segments of the permute mask. Thus, the rejection for claim 5 is improper.

Further, for example, claim 6 recites "generating a new bit pointer using the first result". Note that claim 1 recites "combining the plurality of entries into a first result". In Sazegari, the 6th – 8th bits of a byte of the index register are not generated using the result (32) of the permute instruction.

Further, for example, claim 7 recites "receiving an offset, wherein the plurality of indices are determined using the offset and the plurality of segments of bits". The Office Action relied upon Sazegari (Col. 5, lines 1-20) for this limitation, which is improper. Sazegari (Col. 5, lines 1-20) shows how ***multiple instructions*** may work together to implement the operations as shown in Figure 6. In Sazegari, the indices in the permute mask are at the predetermined locations; and the permute instruction does not use the description in Sazegari (Col. 5, lines 1-20) to determine the indices for looking into the single 32-entry lookup table. Furthermore, receiving a parameter, such as an offset, is completely different from shifting. Thus, the rejection for claim 7 is improper.

Further, for example, claim 8 recites "partitioning look-up memory into the plurality of look-up tables before said looking-up". Sazegari (Col. 2, lines 17-25) shows "dividing a large table into a number of smaller tables". Sazegari does not show the partitioning of "***look-up memory***". Thus, the rejection of claim 8 is improper.

Further, for example, claim 11 recites "the single instruction specifies a total number of entries contained in each of the plurality of look-up tables". In Sazegari, the permute instruction operates on a single 32-entry table, where the total number of entries is a pre-determined number 32. The permute instruction does not specify "a total number of entries contained in each of the plurality of look-up tables". Thus, the rejection is improper.

Further, for example, claim 13 recites "the single instruction specifies a total number of bits used by each entry contained in the plurality of look-up tables". In Sazegari, the permute instruction operates on a single 32-entry table, where each of the table entries has 8 bits which is pre-determined. The permute instruction does not specify "a total number of bits used by each entry contained in the plurality of look-up tables". Thus, the rejection is improper.

Further, for example, claim 17 recites "<u>said combining</u> the plurality of entries <u>comprises</u>: selecting a valid data from the plurality of entries." Note that according to claim 1, "said combining the plurality of entries" is in response to "the microprocessor receiving the single instruction". In Sazegari (Col. 4, lines 63-67), the selection of the final result is performed using at least a further selection instruction, which is clearly not a part of the "combining" operation of the permute instruction. Thus, the rejection of claim 17 and its dependent claims is improper.

Further, for example, claim 26 recites "receiving a first number indicating a position of a last bit of input in the string of bit". In Sazegari, there is no indication that the permute instruction needs the indication of a position of a last bit of input in the string of bit received from the permute mask. Thus, the rejection of claim 26 and its dependent claims is clearly improper.

Further, for example, claim 30 recites "formatting the string of bits into at least one escape data according to the at least one format; and combining the at least one escape data and the first result into a second result". Applicant respectfully submits that Sazegari does not show such a feature.

Since dividing a claim into isolated elements may lead to the improper application of elements of prior art to the claim limitations, as discussed above, Applicant respectfully requests the consideration of each of the claims as a whole, including the interrelations between elements as expressed in the each of the pending claims.

Applicant respectfully submits that the pending claims are patentable over the cited references.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due or credit any overages. Furthermore, if an extension is required, Applicant hereby requests such extension.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 8/9, 2005

Lehua Wang
Reg. No. 48,023

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025-1026
(408) 720-8300